UNITED STATES AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY
Wright-Patterson Air Force Base, Ohio

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

(14) REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
AFIT/GSM/SM/77D-30

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER
(9)

4. TITLE (and Subtitle)
(16) SPARE MEMORY AND TIMING PARAMETERS IN AVIONICS COMPUTER SYSTEM REQUIREMENTS.

5. TYPE OF REPORT & PERIOD COVERED
~~MS Thesis~~ Master's thesis,

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
(11) Gary B. Wigle

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Air Force Institute of Technology (AFIT-EN)
Wright-Patterson AFB, Ohio 45433

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS

12. REPORT DATE
(11) Dec[redacted]77

13. NUMBER OF PAGES
87 (12) 85 P.

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)

15. SECURITY CLASS. (of this report)
Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

APPROVED FOR PUBLIC RELEASE AFR 190-17.

JERRAL F. GUESS, CAPT, USAF
Director of Information

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Avionics Software
Software Maintenance
Software Support

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Avionics computers require continuous software maintenance support during the life cycle of the airborne system. Spare memory and timing capability should be provided with the initial acquisition of the system. Too often, additional capability must be acquired at a later date and at a high cost. Current recommendations for spare capacity vary between 20 and 100 percent. An analysis has been made on 25 computers in 14 Air Force

012 225

LC

airborne systems to determine the growth of software and hardware size to date. The results of this analysis indicate that 100-300 percent spare memory should be provided in avionics computers that process data for navigation, weapons control, radar, electronic warfare, or any other function that has changing mission requirements. Also, only 25 percent spare memory is needed in avionics computers associated with missiles, status monitoring, fault isolation, or similar functions. Not enough data is available to reach any sound conclusions concerning the timing in avionics computers.

SPARE MEMORY AND TIMING PARAMETERS IN AVIONICS COMPUTER SYSTEM REQUIREMENTS

THESIS

AFIT/GSM/SM/77D-30

Gary B. Wigle
Capt USAF

Approved for public release; distribution unlimited

AFIT/GSM/SM/77D-30

SPARE MEMORY AND TIMING PARAMETERS IN AVIONICS COMPUTER SYSTEM REQUIREMENTS

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

by
Gary B. Wigle, B.S.
Capt USAF
Graduate Systems Management

December 1977

Approved for public release; distribution unlimited

Preface

This thesis is the result of my research of spare memory and timing provisions in several Air Force airborne systems. Studies in the past have used cost as a basis for recommending how much spare capability to acquire with a new computer system. This spare capability is wasted, though, unless it is actually needed over the life cycle of the system. Thus, I set out to examine the need for spare capability in avionics computers.

Space precludes citation of all who made this thesis possible. I have included a list of those people who gave of their time and provided me with information in Appendix B.

I with to thank Lieutenant Colonel Larry Taylor (ASD/ENAI) for his assistance in suggesting an area for research. He helped me to formulate the area for research, and then gave me guidance on an approach to use.

I would especially like to thank my thesis advisor, Major Charles McNichols, and my thesis reader, Major Saul Young, for their continued support and direction in this research effort. Major McNichols was most instrumental to the accomplishment of this thesis.

Finally, I would like-to thank my wife, Linda, for her support and encouragement during this very busy time.

Gary B. Wigle

Contents

List of Figures

List of Tables

Abstract

Avionics computers require continuous software maintenance support during the life cycle of the airborne system. Spare memory and timing capability should be provided with the initial acquisition of the system. Too often, additional capability must be acquired at a later date and at a high cost. Current recommendations for spare capacity vary between 20 and 100 percent. An analysis has been made on 25 computers in 14 Air Force airborne systems to determine the growth of software and hardware size to date. The results of this analysis indicate that 100-300 percent spare memory should be provided in avionics computers that process data for navigation, weapons control, radar, electronic warfare, or any other function that has changing mission requirements. Also, only 25 percent spare memory is needed in avionics computers associated with missiles, status monitoring, fault isolation, or similar functions. Not enough data is available to reach any sound conclusions concerning the timing in avionics computers.

SPARE MEMORY AND TIMING PARAMETERS IN AVIONICS COMPUTER SYSTEM REQUIREMENTS

I Introduction

The use of software in avionics systems began in the late 1950s. At that time, a digital computer was first used with the MA-1 fire control system in the F-106 (Ref 3:50). During the twenty years since that time, avionics software has become a multi-million dollar annual business just within the Air Force. Avionics software has found applications in nearly every function aboard an aircraft. It is now found in systems aboard the B-52, C-141, A-7, AC-130E, C-5A, F-111, F-15, B-1, and numerous others (Ref 14:12). Many problems were encountered, though, during this rapid growth.

Between 1972 and 1975, a number of conferences were held with the objective of analyzing the problems associated with software management in the military services. These conferences examined problems associated with all types of software and made numerous recommendations. One such study, often referred to simply as Electronics-X, makes a recommendation to "select a processor of adequate size to permit underutilizing the computer; write highly modular programs; emphasize structure and overall efficiency rather than hardware efficiency alone" (Ref 12:304). The Electronics-X study recognized that one of the major sources of excessive software costs in conventional systems is selecting a central processor too small, with consequent overutilization of the computer and use of programming practices which decrease software reliability and increase software maintenance cost (Ref: 12:303).

Problem Definition

In many of the study group reports, the key recommendation involves standardization. The incentive to standardize is strong when viewed as a way to reduce what have become overwhelming software costs. As an example of this drive toward standardization, an attempt has been made to standardize the requirements for spare memory and spare timing capacity; that is, to standardize the underutilization required for a delivered operational avionics computer system.

The memory capacity of avionics computers is usually within a range varying from a few thousand words up to 64,000 words or more. Since most requirements can be satisfied with data words of 15 or 16 bits. while certain other functions require 24 to 32 bits, a trend has been established to use a 16-bit word length, or a 32-bit word length with full word and half-word instruction and data capability (Ref 24:139).

The speed usually refers to the computational capability of a processor. It is stated in terms of the millions of instructions per second (MIPS) that a processor is capable of executing. This capability is often a measured average of execution times of certain mixes of various instructions. Processor speed may also be stated in terms of thousands of operations per second (KOPS) (Ref 24:139).

In Air Force use, timing is not synonymous with speed of the computer. This author found that personnel contacted at several Air Logistics Centers (ALCs) assume timing to mean different things. The most prevalent concept among these personnel is that timing refers to the cycle time of the programs. This cycle time is the length of time it takes for the entire program to execute. It varies with the amount of data to be processed and the number of times each subroutine is to be executed. The timing capacity is the maximum time allowed in the design requirements for the cycling of the programs. Worst case conditions are usually set up for determining whether or not the programs meet the timing requirements (Ref 30).

Within the Air Force, Aeronautical Systems Division (ASD) has presented its attempt to standardize these two parameters (spare memory and timing capacities) in EXHIBIT ASD/ENAIA 76-1. It states, "A system capacity (minimum 25 percent timing and 40 percent memory) to provide growth capability consistent with the anticipated level of computer program support during the life of the system will be included" (Ref 6:7). These requirements apply to either spare memory actually provided at the time of acquisition, or the capability of the central processor to address further memory when an add-on design is used (Ref 30).

The problem here is that the two figures stated, 25 percent for timing and 40 percent for memory, have no solid basis. They were chosen for reasons unknown at the present time. The current staff would like some basis for the figures chosen to be used for required underutilization. That is the purpose of this report, to determine appropriate values to be used for spare memory and spare timing capacity requirements for operational avionics computer systems at delivery.

Objectives

The specific objectives of this report are twofold. The first is to propose and defend a standard optimum software parameter choice for each of the two parameters discussed. The second objective is to attempt to identify feasible parameter range choices for various avionics software functions. It may be that one parameter will not be sufficient for all software functions. If this is the case, then the reasons for not using one parameter should be made clear.

Scope

This research effort deals only with the two parameters noted and excludes all others from consideration. The value recommended for each parameter is derived from a study of actual growth in past and present Air Force airborne systems. This study is not based on a theoretical cost trade-off analysis. Dr. Boehm of the RAND Corporation has already completed such a study, and his results will be used in this report.

It may be even more important to note what is not within the scope of this report. No attempt is made to predict the effects future hardware developments will have on the parameter problem. When studying the changes made to a software system, no attempt is made to analyze why a change was made or if it was really necessary.

General Approach

The objectives of the research were accomplished primarily through an historical study of digital avionics systems. This study determined the actual memory and timing growth thus far in various avionics software systems. It attempted to answer questions such as: Has all spare memory been used? Were most changes made during the early years or the latter years of the life of the system? Which is most critical, spare timing or memory? Are newer systems results substantially different from older systems results? Is a trend developing?

The systems have been subdivided into various functions to see if results differ to a large degree between them. A parameter range was attempted for each of these functional areas. Some of the functions examined were electronic warfare, command and control, navigation, radar, and weapons guidance. It is logical to assume that some functions have many more changes required than others. Once the breakdown analysis was completed, a determination of whether or not a single parameter can be used for all functional areas was made.

Overview of the Thesis

Chapter II gives a history of software as the Air Force transitioned from a hardware first philosophy to a software first philosophy. Avionics software, software errors, software maintenance, and the cost of software are discussed in detail. Chapter III is a presentation and discussion of the data gathered on a number of Air Force airborne systems. Data concerning memory and timing parameters are tabulated for each computer aboard these systems. Chapter IV contains an analysis of the data. Problems encountered and limitations of the data are noted. A comparison is made between the actual memory and timing needs and the stated requirements in the ASD/ENAIA EXHIBIT. Chapter V includes a list of conclusions and recommendations concerning spare memory and timing parameters for acquisition of future avionics computers.

II Background and History of the Problem

This chapter contains an overview of the problem. It includes a discussion of how the emphasis has changed from hardware to software in the development of computer systems in the Air Force. Software is defined and the cost of software is illustrated. The discussion is then narrowed down to avionics software. Reliability of software is defined and software maintenance explained. Some current thoughts on provision of excess capacity and some future concerns conclude the chapter.

Hardware First Philosophy

The Air Force began its entry into the computer technology field in the 1950s with a number of different systems, some of which are still in operation. One is the system that provides air defense for the United States and is called the Semi-Automatic Ground Environment (SAGE) system. Another is the fire control systcm ᵒⁿ ʟhe F-106. These and other systems have provided much information about problems encountered with software maintenance during the life of a system.

The computer techrology field has grown quickly in the last twenty year:. For those interested, a detailed history of this growth is provided in a RAND study titled Air Force Command and Control Information Processing in the 1980s (Ref 2:35-49). In the early stages, hardware capability was the limiting factor in system development. As pointed out in a study prepared for the Office of Naval Research, "In many projects, especially real time systems, the software effort has to wait until the hardware is procured, or at least until the selection is made. Then the programs are written under the hardware constraints" (Ref 23:2). In many situations, there was no alternative because avionics computers had to satisfy restrictions on physical size. As technology has progressed, the policy of developing software to fit hardware requirements has carried forward, even though improved technology has made the physical size problem less important.

This hardware first philosophy is an important hurdle to overcome. In past procurements, personnel tended to be preoccupied with hardware requirements, such as weight and size. They were not too concerned about the resulting software ramifications. This was largely because the project engineer could easily identify with hard characteristics. He could see, feel, and kick hardware. It was difficult to identify with software. As a result, in many avionics procurements, arbitrary and restrictive constraints were applied (in the name of "cost effectiveness") with little or no concern for the resulting high costs of software (Ref 16:101).

A typical example illustrates this philosophy. During the early development cycle of one aircraft, core memory estimates for performing all functions in one computer ran as high as 24,000 (16-bit) words. The airborne computer used contained only 16,896 words. The programmers were forced to use tricks to keep from exceeding the memory size limits. Such practices resulted in difficult and expensive software maintenance during its operational life (Ref 35:101-102). The Air Force Logistics Command (AFLC) has recognized this problem and noted that the present acquisition policy leads to a serious problem with spare computer processing capacity. If a program is estimated to have 32,000 words, a computer with a 32,500 word capacity is obtained (Ref 29:46).

Hardward development has progressed extremely fast and airborne computers currently in production or in development are of two major types:

1. Large, high-performance computers for airborne command and control system support and for the airborne processing of data received from sensor platforms.
2. Avionics computers used in aircraft for various on-board processing functions.

The memory capacity of airborne computers has grown from a range of 1000 to 4000 words in early years to 131,000 to 262,000 words in the 1970s (Ref 16:83,90). These values will, no doubt, increase even more in the years to come.

Software

Software is not as easy to deal with as hardware. The first problem is defining what software is. It is not something one can feel and touch like hardware. It has intangible properties. Some would define software to include all computer programs and their documentation (Ref 12:296). This is the most widely accepted concept. This report will define software as all computer programs and data used by them. The computer programs are the sets of instructions to be executed, and the data is that information processed by them. Software is not to be confused with the program listings which are considered documentation. As a legal contract is an agreement often "represented" by a piece of paper, the computer programs are machine logic through programmable instructions represented by program listings. The real software, in itself, has no physical properties.

Cost of Software

The importance of software has been emphasized in recent years for one reason: costs! Where hardware once dominated costs, software now does by a wide margin. The trend is illustrated in Figure 1.

In 1960, software costs represented only about 25 percent of the Air Force budget for Electronic Data Processing (EDP) while in 1973, software costs represented 75-85 percent of the USAF budget for EDP (Ref 28:1). It is estimated that by the 1980s, software costs will be about 90 percent of the budget for EDP, as hardware costs continue to go down and people costs continue to go up (Ref 29:29).

The percentages by themselves do not say much without the amount of dollars known. In 1972, the Air Force spent

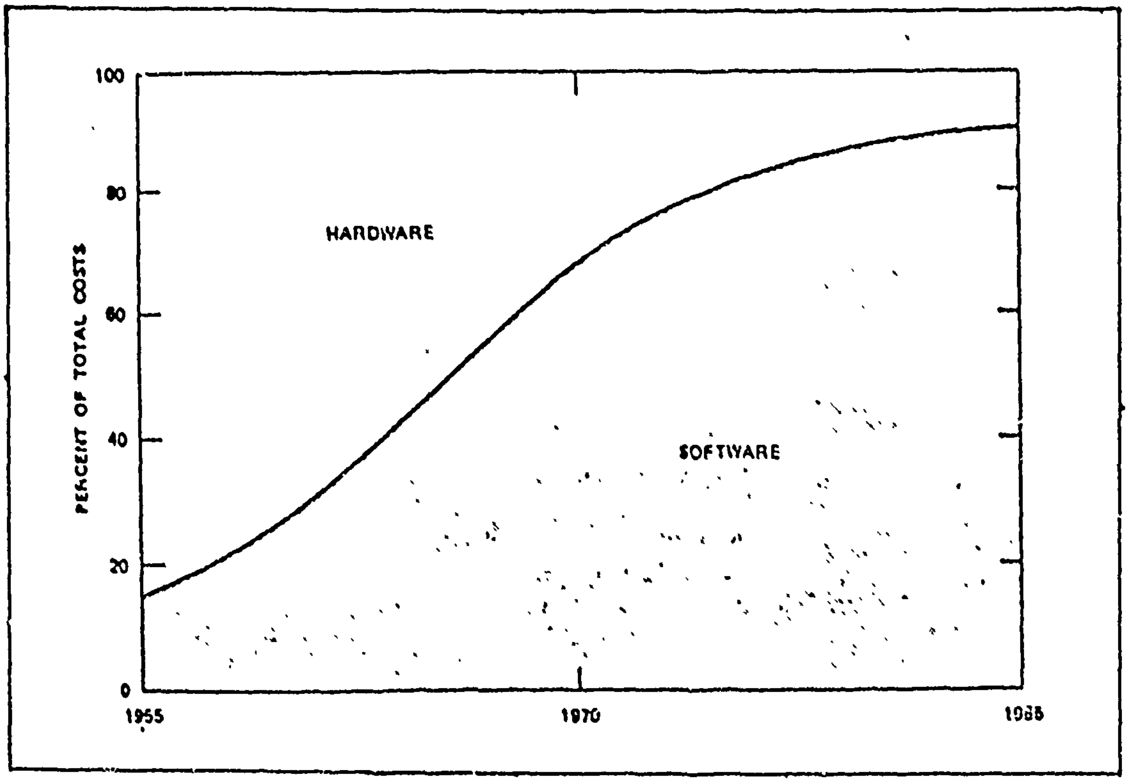

Figure 1. Hardware/Software Cost Trends (Ref 27:28)

between $1 billion and $1.5 billion on software; that is, computer programs and their associated documentation (Ref 12:37). At that time, the costs represented about 4-5 percent of the total Air Force budget (Ref 34:2). By 1975, this figure had grown to about $2.5 billion (Ref 36:2).

These costs, of course, are for all software in the Air Force. Since this report deals with avionics computers, it might be interesting to note what the costs are for this specific application. Dr. Barry W. Boehm of the RAND Corporation showed that his rough estimates for the distribution of USAF software costs by application were: management information systems, 33 percent; scientific and engineering, 23 percent; command and control and intelligence, 21 percent;

logistics and maintenance, 13 percent; and avionics, 10 percent (Ref 27:13). Further, of this 10 percent of USAF software costs, one avionics project demonstrated that only 15 to 25 percent of the costs of that project were for the Operational Flight Programs (OFPs). The remaining costs were for support programs and system validation (Ref 4:75). This would indicate that possibly 1.5 to 2.5 percent of Air Force EDP costs are for the actual OFPs. This still translates to between $35 million and $65 million annually.

Note that the costs above are for the development of avionics software and not for its maintenance. Because of trade-offs during development between speed of creation and cost of maintenance, current Air Force avionics software costs are about $75 per instruction for development and up to $4000 per instruction for maintenance (Ref 27:14), a rather interesting range of costs. It is now estimated that current Air Force maintenance costs on a software system are roughly equal to the initial development costs, during the life cycle of the system (Ref 34:5). This is the reason for the present emphasis on life cycle costs of software during the initial stages of procurement.

Avionics Software

Operational avionics software is all software that resides in, or is a part of, a functional system or subsystem. This software consists of three major classes:

operational flight software, automatic test equipment (ATE) software, and crew training simulator software (Ref 23:4-5).

The operational flight software includes all computer programs executed in the airborne system. These programs could include an offensive flight program, a defensive flight program, software in the central air data computer, mission software (such as that used in the Airborne Warning and Control System), and programs for data processing in any other airborne processor. More than one computer may be involved, but the total of all software items for a given aircraft system is referred to as the operational flight software for the system, regardless of whether or not the system is integrated (Ref 23:5).

ATE software includes the computer programs used to control test operations and the procedures required to test various hardware systems and subsystems. Basically, the programs generate test stimuli and measure the response of the system or subsystem, and then compare these responses with predetermined acceptance parameters. The ATE software and equipment is used to support test activities in both depot and field environments (Ref 23:5).

Crew training simulator software provides simulation of the weapon system performance and operating characteristics, simulates the environment in which the weapon system operates, and provides for instructor control (Ref 23:5). As an example, to train F-15 pilots in a more economical manner, flight simulators are used. The crew training simulation software is the software used to operate these flight simulators. This software is procured at the same time the operational flight software is procured.

This report is concerned with the operational flight software. The airborne avionics software starts its development with the basic operational requirements. The various avionics functions are defined and then the software requirements are determined. These requirements then dictate the computer capability necessary; i.e., the software memory and time requirements, and therefore the computer speed and memory size. The computer is designed or selected and the software requirements are translated into basic functional flow charts (Ref 19:2). Some of the functions for which software is developed are navigation, weapons delivery computations, control and display processing, status monitoring and fault isolation, and electronic warfare (Ref 24:80-81).

A typical software system life cycle is illustrated in Figure 2. The phases of this life cycle can be grouped together as requirements analysis, software production, and system operation (Ref 2:10-11).

Software Maintenance

During the operational phase, problems will arise and changes will have to be made to the software. This is referred to as software maintenance. It is different from

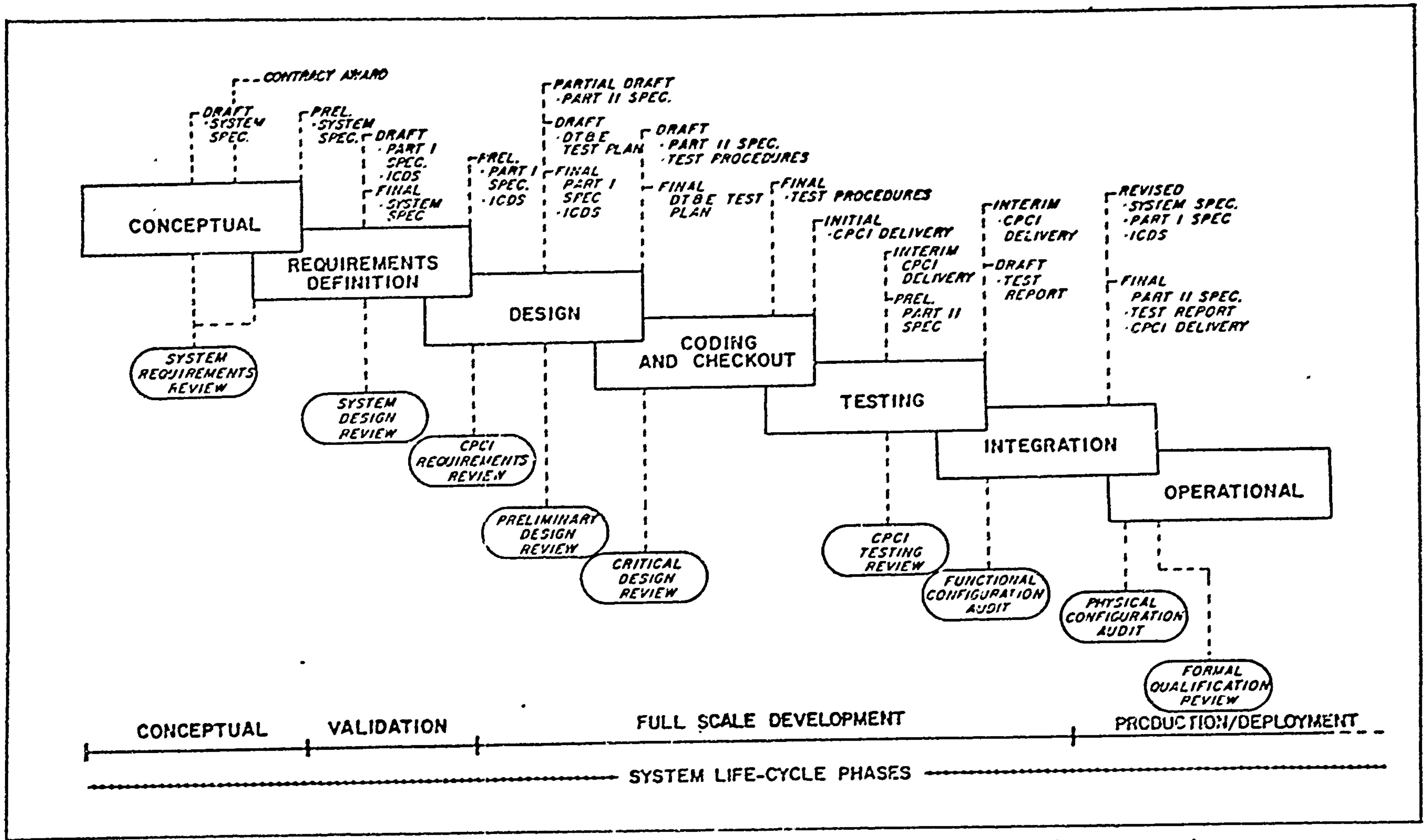

Figure 2. Software Life-Cycle Phases and Major Milestones (Ref 22:8)

hardware maintenance in that software does not wear out or degrade in performance with age. Software maintenance is modification or change in the program instructions. It may be required for any of the reasons discussed in the next paragraph. If the required modification is large, the software maintenance may require another development cycle (Ref 23:39).

Software maintenance may be required because of one or more of four very valid reasons: a change is required in the mission, an error is found which must be corrected, the program must be optimized in size or speed, or the program must be adapted to a hardware change. If the software is to do something different, change is required. This change is considered maintenance action if the system is in operational use. Nearly every computer program is considered to contain some errors which may remain undetected for years. But when an error is found, common sense says that a correction should be made, resulting in more software maintenance. When memory capacity becomes critical or when execution time becomes too slow for the real time requirements, optimization of the program may be necessary. This may be required more than once if hardware constraints prevent memory add-on. Program optimization is in a sense a "technical" change, in that the program still performs the same functions and nothing new is added. In addition, changing hardware continually requires changes in software, sometimes for increased capaoility and sometimes to compensate for changed hardware characteristics (Ref 14:15).

Software maintenance can be accomplished by three different approaches. In the past, frequently, a contracted-only effort was established, whereby the Air Force assigned the technical job of software modification, documentation, and testing to a software contractor. With this approach, the Air Force was not usually concerned with the spare capabilities of the computer to any extent. The other extreme is a completely in-house approach, such as that ultimately used with the SAGE system after 1967. This approach is receiving more and more emphasis and is the drive behind the growing concern for the spare capabilities of a computer system. The third approach is a combination of the first two approaches. Minor changes and updates would be made by in-house personnel, while major changes would be made by a contracted effort. This last approach may be dominating the current Air Force software maintenance scene (Ref 36:7-8).

It must be emphasized that software maintenance is a major component of total software expenditure. For example, a Hoskyns survey in Great Britain studied 905 installations there and found that almost 40 percent of the software effort in Great Britain is software maintenance. This estimate is comparable to the avionics software effort within the Air Force (Ref 27:29-30).

Software Errors

Three of the four reasons for changes discussed previously can be, to some extent, controlled. The fourth one, software errors, cannot easily be controlled. A program cannot be expected to be error free. Errors may be caused by a mistake in coding. They may be caused by the incompatibility of the program and the computer hardware. Errors can occur due to truncations or imprecision in the calculations. More resources may be requested by the program than can be supplied. Also, errors may be caused by complex timing problems (Ref 28:6). Logicon Inc. estimates that operational flight software will have approximately a five percent error rate in the original code (Ref 36:49-50).

Extensive testing and validation is often used to make the program as error free as possible before it enters the operational phase. But even testing has its limitations. The largest software program that has been absolutely proven to be error free has some 400 instructions (Ref 34:3). Testing and validation must demonstrate that hardware and software together produce the desired results (Ref 4:97). But it must be remembered that program testing can be used to show the presence of "bugs," but never to show their absence (Ref 5:5).

Ideally, software validation involves checking all possible logical paths through a program. Dr. Barry Boehm, of the RAND Corporation, illustrates the complexity of this

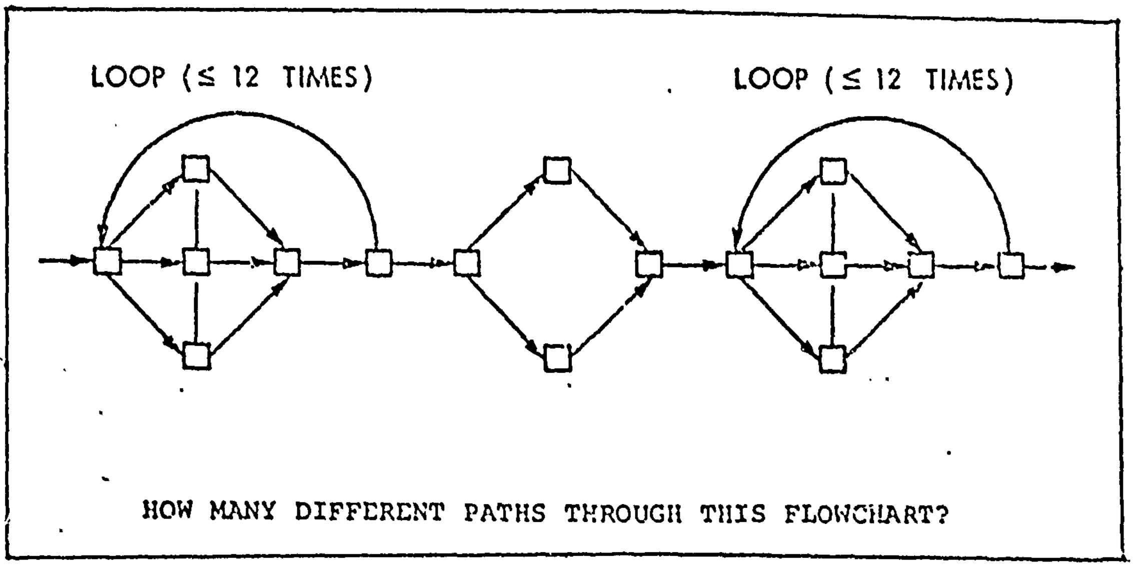

Figure 3. Simple Program Flow Chart (Ref 8:24)

task. Figure 3 shows a rather simple program flow chart But even through this simple flow chart, the number of different paths is about ten to the twentieth. If one had a computer that could check out one path per nanosecond (10^{-9} second), and had started to check out the program at the beginning of the Christian era (1 A.D.), the job would be about half done at the present time (Ref 8:23-25).

Figure 4 shows the relation of errors found to testing effort in the development of software for several medium-sized programs at the System Development Corporation. The figure shows that the last 50 percent of the testing effort found only 15-25 percent of remaining errors (Ref 2:60).

When an error is found, a change must be made. This sounds pretty simple, but how many more errors will these corrections produce? One study indicated that 19 percent

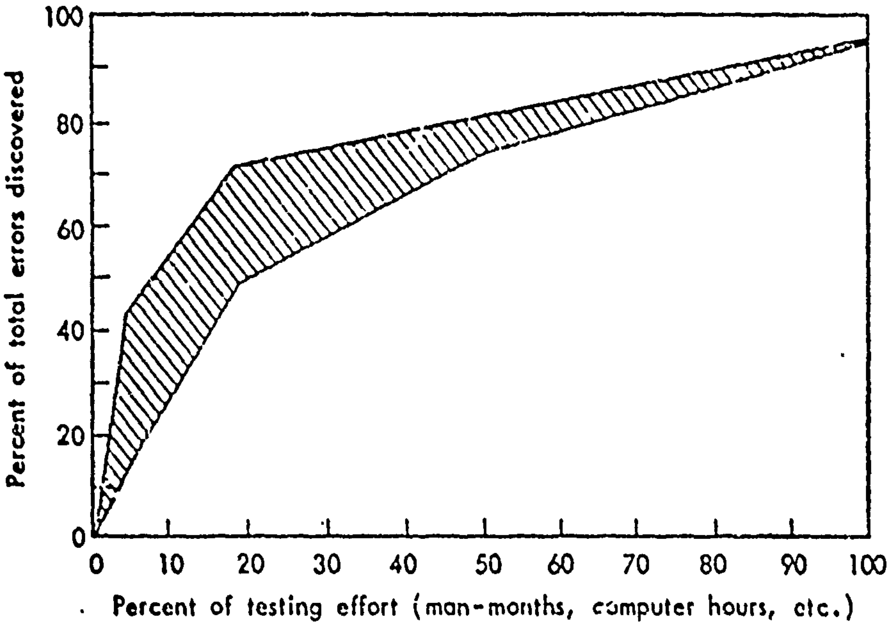

Figure 4. Relation of Errors Found to Testing Effort in Software Development (Ref 2:61)

of the errors in a set of programs resulted from <u>unexpected side effects to changes</u> (Ref 28:9). Figure 5 shows the relationship between successful first runs of corrections and the number of statements modified. Even after a small modification, the chance of a successful first run is, at best, about 50 percent (Ref 8:29). A classic example of this is found in the SAGE system. The correction of a one-word error required three official program corrections issued by the SAGE Programming Agency before it operated correctly.

Because of the errors generated by software corrections, the error rate of a system will be similar to that

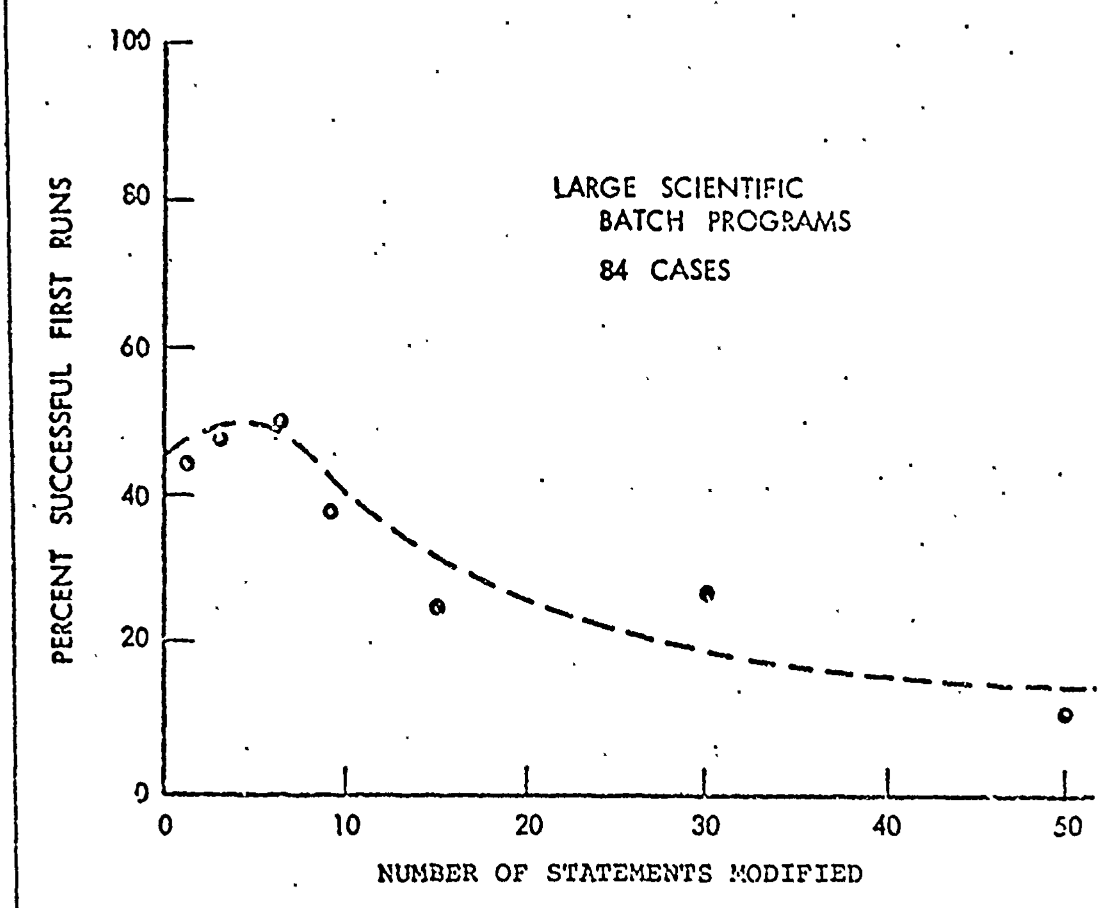

Figure.5. Relationship Between Successful First Runs of Corrections and the Number of Statements Modified (Ref 8:28)

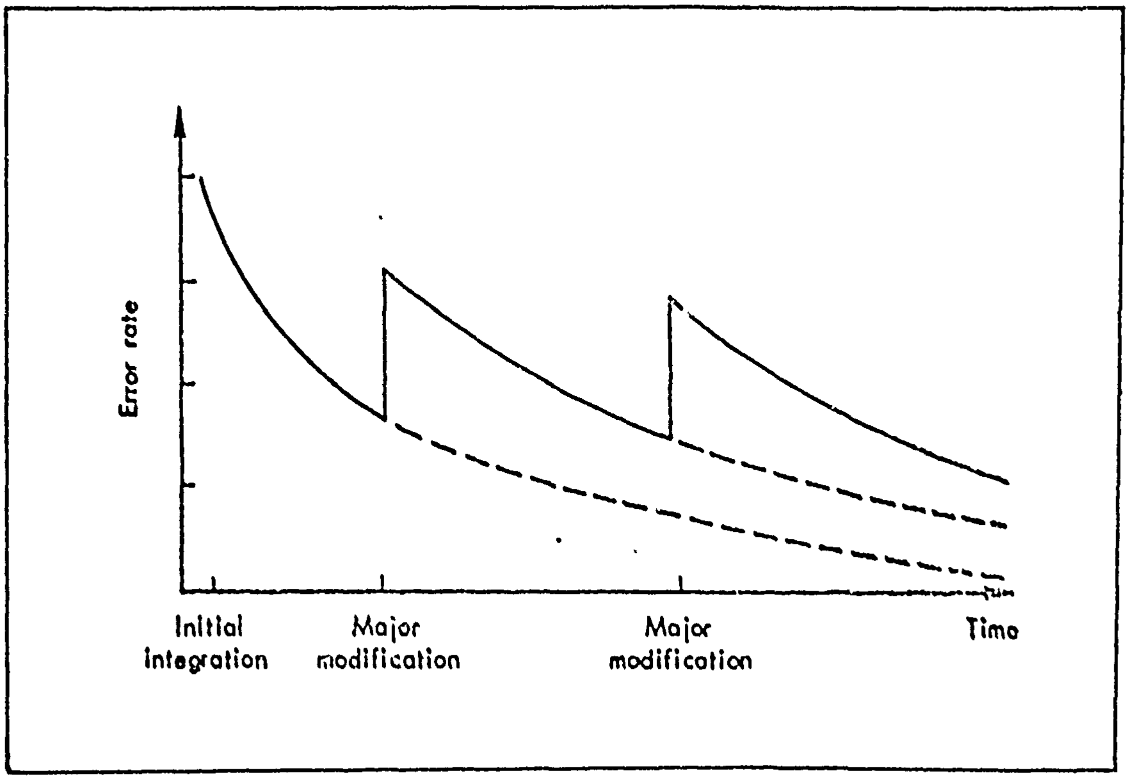

Figure 6. Harmful Effect of Modifications on System Stability (Ref 2:69)

shown in Figure 6. Software maintenance will be continuously required throughout the life of a system. These factors must be considered when system requirements for timing and memory sizing are determined.

Software First Philosophy

Because of the problems involved with software maintenance, failure to insure adequate computer sizing in terms of speed and memory results in numerous difficulties. System operation may degrade to the point of decreased mission effectiveness, and growth capability for future software may be nonexistent (Ref 26:30).

These hardware constraints affect productivity. To squeeze the program down may require tightening of the coding and perhaps using questionable tricks to shorten the program. The RAND Corporation points out that when a program must fit into a restricted memory or meet strict timing requirements, much more effort is needed to achieve software efficiency. When software requirements have pushed hardware capacity, gains in program efficiency have been bought at the cost of logical complexity. Machine language has been used instead of Higher Order Languages (HOLs); multiple data elements have had to be packed into single memory cells; and, tricky programming techniques (such as multipurpose flags and counters, reusable sections of code, and the like) have had to be used (Refs 2:82; 12:300). It has been shown that procuring extra memory and speed during the development phase can reduce software cost by more than the additional hardware cost (Ref 31:517).

A sample of recommendations from several studies and recent workshops illustrate this trend:

> Select a processor of adequate size to permit underutilizing the computer; write highly modular programs; emphasize structure and overall efficiency rather than hardware efficiency alone (Ref 10:2-5).
>
> Hardware capacity be specified with adequate allowance for a safety factor to reduce the difficulties of programming (Ref 10:4-2).

> Planning for software support (supportability) must begin during the conceptual phase of system design (Ref 10:6-2).
>
> The Air Force should consider weapon system computer hardward and its related software as an integral problem - decisions regarding one should be made with full recognition of the other (Ref 10:6-3).
>
> Software which performs the required functions is most useful when it is sufficiently flexible or changeable so that quick modifications can meet urgent mission requirements (Ref 10:9-9).
>
> The software-first concept seems sufficiently promising to merit more detailed studies of its ramifications and alternatives, followed by exploratory or advanced development if appropriate (Ref 10:5-5).

Excess Capacity

Cost emphasis has shifted from hardware to software. In the early 1970s, it was shown that software costs were 2 to 3 times hardware costs. One study estimates that the cost ratio of software to hardware will be approximately 9 to 1 by 1985 (Ref 10:6-3). Another source believes that the ratio could go as high as 10 to 1 (Ref 5:3). Dr. Boehm (of RAND) developed the data shown in Figure 7. This figure shows how the total data processing system cost varies with the amount of excess capacity procured for various estimates of the ratio of ideal software-to-hardware costs for the system. These ideal software costs are those that would be incurred without any considerations of straining hardware capacity (Ref 8:7).

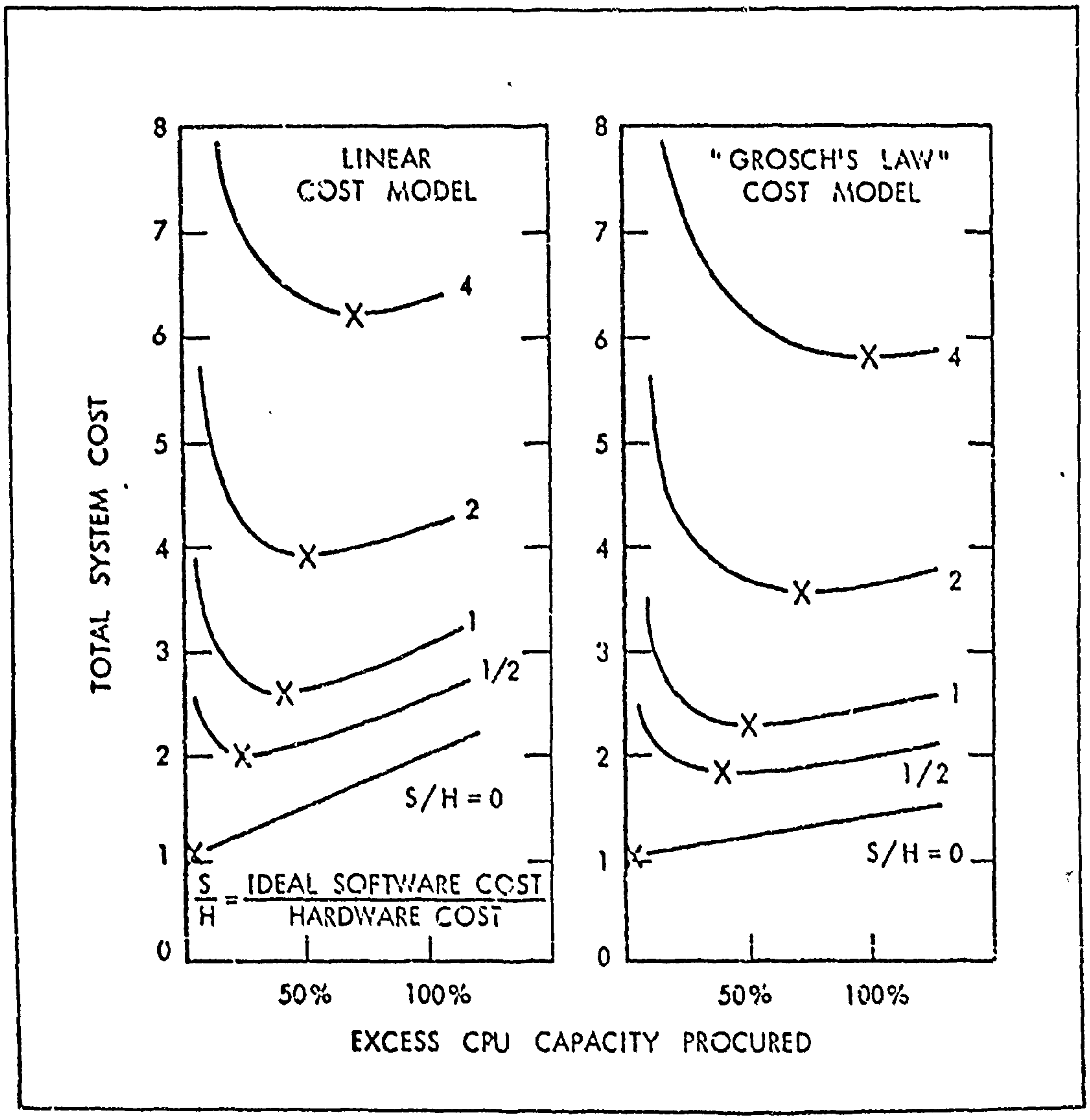

Figure 7. On-Board Computing, Total System Costs (Ref 8:8)

Two models of hardware cost are used in Boehm's study: the linear model assumes that cost increases linearly with increased capacity; and the "Grosch's Law" model assumes that cost increases as the square root of capacity. Both models show similar trends. One point is that the more the ratio of software-to-hardware cost increases, the more excess capacity one should procure to minimize the total cost. One can see from Figure 7 that one should acquire quite a large amount of excess capacity if the ratio does go as high as 9 or 10 to 1. A second point is that the overall system cost is generally minimized by procuring computer hardware with at least 50 percent to 100 percent more capacity than is absolutely necessary (Ref 8:7).

Dr. Boehm did another study of 34 airborne and spaceborne software projects. He studied the relative programming cost per instruction as the hardware capacity approached 100 percent utilization. The results of this study are shown in Figure 8. This study demonstrated that when hardware capacity became 75-80 percent utilized, the cost for programming rose sharply. Again, this led to his 50-100 percent or more recommendation for excess capacity (Ref 8:4-5). Two other studies specifically used Dr. Boehm's results to recommend a 50 percent excess capacity when procuring a computer system (Refs 12:300; 1:29). This study is the most referenced recent study of software costs.

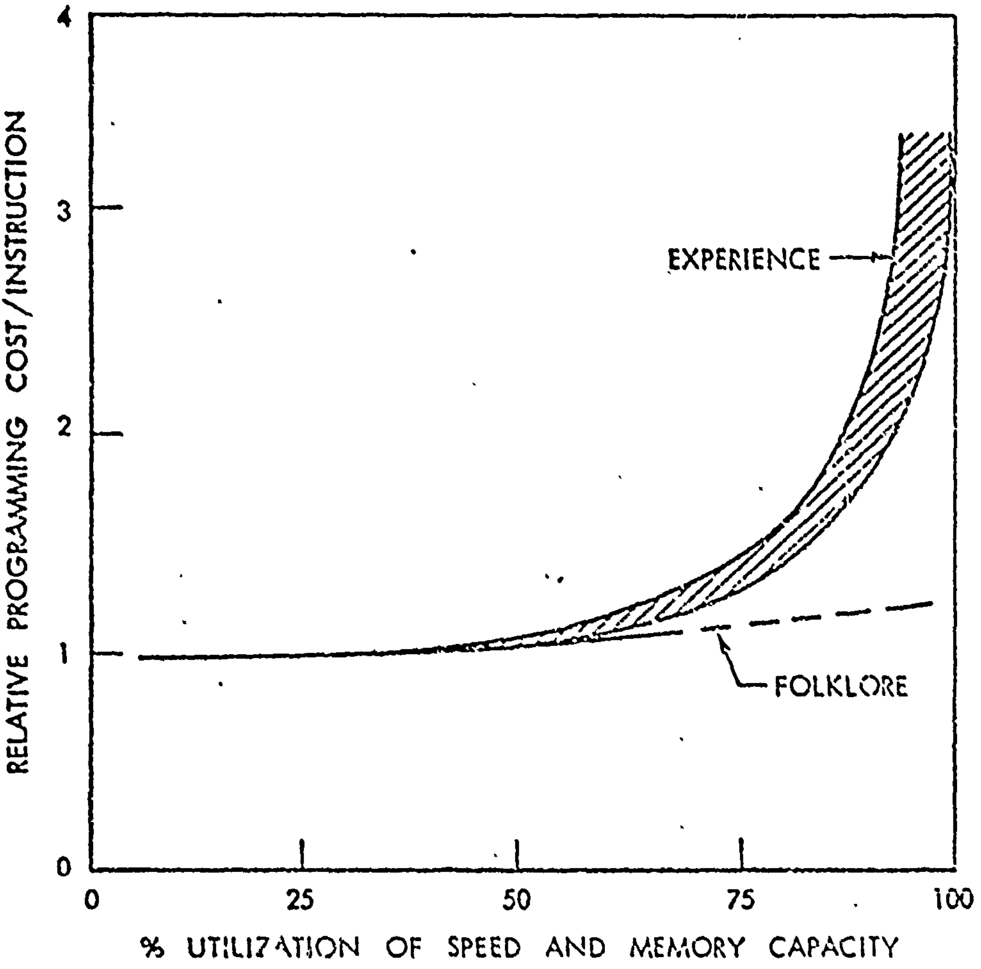

Figure 8. On-Board Computing,
Software Costs (Ref 8:4)

There have been other values recommended in other reports. One report stated that a surplus of 25 to 30 percent is considered representative (Ref 11:201). That report did not explain, though, how that estimate was arrived at or exactly what it was representative of. Also, the Navy established a policy of providing a 20 percent spare memory and timing capacity at system delivery. It noted that spare timing capacity may be even more critical than spare memory because it is more difficult to expand (Ref 18:2-9). And, as was stated in Chapter I, Aeronautical Systems Division has said that it wants 40 percent spare memory and 25 percent spare timing capability when a system is delivered as operational (Ref 6:7). The values recommended in these reports appear to vary within a wide range.

In the studies mentioned previously, there is frequently no distinction made in values for spare memory and spare timing capacity. One figure is usually recommended for both. USAF/ASD appears to be one of a few organizations that think different values are appropriate.

Future Concerns

About the only thing one can say about future concerns is that there is little agreement among those in the computer technology field. One study concludes that airborne computers of sufficient speed, size, and hardness for the 1985 era will not exist without a dedicated research and development effort (Ref 9:8). Another study states that future military hardware will satisfy environmental requirements for airborne systems (Ref 2:93).

One study states that the Air Force must standardize its software, and one of the ways would be through the use of Higher Order Languages (HOLs). It is estimated that a good HOL would use only 10 percent more memory than an assembly language (Ref 11:200). Yet, another study concludes that while it is felt that future systems should tend to standardization, the nature of present systems does not lend itself to it (Ref 22:8-3). The increased use of firmware in the future will also ease the burden on software and may aid standardization.

Another study projects that airborne computers in the 1980s will have an add-on capability. Therefore, it says that there will then be no technical reason for ordering extra hardware capacity to accommodate possible sizing errors (Ref 16:10). But, as was stated in Chapter I, add-on capability is included within the scope of this report. Even with an add-on feature, the design of the computer system must include the capability of the central processor to address the anticipated add-on memory. Again, how much add-on memory should be anticipated? Excess capacity is still required with the add-on capacity.

Excess capacity, whether through actual spare capability at delivery or through add-on capability later, will be required. The only question concerns how much to require.

III Memory and Timing Parameters for Various Airborne Systems

Data gathered on 25 computers in 14 Air Force airborne systems are presented in this chapter. The basic information is tabulated for each system, but this tabulated information alone is misleading. Along with each table is a narrative which describes problems encountered in maintaining the software, rewrites for optimization or deletions, and expansions or planned expansions of capability. The narrative and tabulated information must be used together to get a complete idea of what has happened to each system. A short description of how the data was obtained precedes the presentation of the data.

Methodology for Data Collection

Since this study was to be based on past and present operational airborne systems, the first step was to contact those personnel in Air Force Logistics Command (AFLC) responsible for software maintenance, AFLC/LOAK at Wright-Patterson Air Force Base. Software maintenance for OFPs, ATE software, and simulation software is controlled by this office. One individual monitors software maintenance for each type of software. At present, Mr. Mark van den Broek is responsible for monitoring software maintenance of the OFPs.

Each system is assigned to an Air Logistics Center (ALC) for actual software maintenance. AFLC/LOAK was able to provide information about which airborne systems are maintained by each ALC. Warner Robins ALC is responsible for Radar Warning and Electronic Warfare systems in general. The F-106 and the C-5A are maintained by the San Antonio ALC. Oklahoma City ALC maintains the software for the A-7D, E-3A, SRAM Missile, and the SRAM Missile Carrier (the B-52 or the FB-111). Ogden ALC is responsible for the F-4E, RF-4C, Minute Man I, Minute Man II, Minute Man III, and Titan missiles. And, the Sacramento ALC is responsible for the software support for the F-111 series aircraft.

The office of responsibility at each ALC (MMEC) was initially contacted by telephone. A list of all personnel interviewed is included in Appendix B. Some of the information was obtained over the telephone, while other ALCs asked that a request for the information be mailed to them. A short form for the collection of the basic data was created to be used in either situation. A sample of this form is included in Appendix C.

This author also contacted some of the System Program Offices (SPOs) at Wright-Patterson Air Force Base. The software for the B-1 and the F-16 were not complete at this time. The A-10 has no on-board software, since the computers on-board use firmware. Data was obtained on the F-15 software through YFEA at the F-15 SPO.

The data collected in this research effort is presented in the remainder of this chapter.

The F-106

The summarized information for the F-106 is shown in Table I. But the information in the table does not show the problems encountered with memory and timing in the past.

The original computer in the F-106 prior to 1969 contained a memory with only 8,000 words. The size of the original program is not known but was something less than 8,000 words, obviously. The original program quickly grew to capacity. In 1969 the HAC-204 was acquired with 34,000 words in memory. The present software occupies about 30,000 words. This means that during the life of the F-106, the growth of the software has been over 300 percent (Ref 17)!

Timing became a problem in 1972 when an attempt was made to digitize the Automatic Flight Control System.

Table I
Computer Data for the F-106 (2nd Computer)

Computer	HAC-204 (IRAM)
Function(s)	Navigation, Weapons Control, Data Link, Tactics Decisioning
Acquisition Date	1969
Memory Size	34,000
Bits/Word	16
Program Size at Acquisition	about 12,000
Present Program Size	about 30,000
Original Cycle Time	Not available
Present Cycle Time	Not available

(From Ref 17)

The problem was corrected through software techniques. The personnel at the San Antonio ALC do not consider timing to be a problem. That is, timing can almost always be handled through programming techniques. Once the computer has been acquired, seldom can its speed be changed. The software maintenance must be done within that hardware constraint (Ref 17).

The C-5A

The programs for the two navigation computers aboard the C-5A have grown to capacity. The data for the C-5A computers is contained in Table II. This software expansion represents a 100 percent growth in the primary computer and about a 54 percent growth in the auxiliary computer. What these figures do not show are the number of program rewrites for optimization. In the last three years, there have been five rewrites to optimize small parts of the programs to obtain more memory. Consequently, if memory had been available, the software growth would have been even greater than the amount just stated. Timing has not been considered a problem for the two navigation computers (Ref 17).

The third computer aboard the C-5A is unique in many ways. It is the MADAR computer and performs status monitoring and fault isòlation. The system was designed with no anticipatation of additional requirements in the future. It included all requirements from the beginning. The

Table II
Computer Data for the C-5A

Computer	Northrop NDC-1060
Function(s)	Navigation
Acquisition Date	Not available
Memory Size	12,000 RAM
Bits/Word	28
Program Size at Acquisition	about 6,000
Present Program Size	almost 12,000 (98%)
Original Cycle Time	Not available
Present Cycle Time	Not available
Computer	Northrop NDC-1060
Function(s)	Navigation (aux)
Acquisition Date	Not available
Memory Size	8,000 RAM
Bits/Word	28
Program Size at Acquisition	5,200
Present Program Size	almost 8,000 (98%)
Original Cycle Time	Not available
Present Cycle Time	Not available
Computer	Northrop NDC-1060
Function(s)	MADAR (status monitoring)
Acquisition Date	Not available
Memory Size	16,000 RAM
Bits/Word	28
Program Size at Acquisition	almost 16,000
Present Program Size	almost 16,000
Original Cycle Time	Not available
Present Cycle Time	Not available

(From Ref 17)

result of that design was software that occupied nearly 100 percent of the available memory. There was some spare memory provided for correction of software errors, but no exact figures are available. Because of the function of this computer, only minor changes have been made with essentially negligible growth to the software. With little growth in the software, timing has not been a problem (Ref 17).

The A-7D

Table III presents the data for the A-7D computer. The software growth has been about 25 percent, thus far. As capacity was approached, the LORAN program was deleted along with some other smaller functions. These deletions made available about 800 more words. The Navy is developing a TC-2A with a 32,000 word memory (expandable to 64,000

Table III
Computer Data for the A-7D

Computer	IBM TC-2
Function(s)	Navigation, Weapons Delivery
Acquisition Date	July 1975
Memory Size	16,000
Bits/Word	16
Program Size at Acquisition	13,000
Present Program Size	approaching 16,000
Original Cycle Time	200 msec complete loop, 5 cycle program.
Present Cycle Time	Same as above

(From Ref 32)

words) and a much faster speed. This change would allow for an anticipated growth of as much as 300 percent in the software. It would also reduce some of the present timing restraints (Ref 32).

As core usage is approaching capacity, the personnel at the Oklahoma City ALC see three alternatives:

1. Remove various capabilities on a priority basis to make room for new material.
2. Produce mission-oriented program tapes, which would reduce the capability and flexibility of each aircraft.
3. Expand core to 32,000 words by replacing the TC-2 with the TC-2A.

Alternative 3 will cost $100,000 for each aircraft to replace the TC-2 with the TC-2A. For 408 aircraft, this will require an expenditure of $40.8 million (Ref 32).

The F-4E and RF-4C

The data in Tables IV and V show that the original memory size of the ARN-101 computer was 32,000 words. It has already been expanded to 48,000 words, and this system is not yet operational! It is only now being flight tested. A request has been submitted to expand the memory to 64,000 words for future changes to the software. One of the design requirements for the ARN-101 computer system is that there be a 20 percent spare timing capability and a 20 percent spare memory capacity when the system becomes operational (Ref 25).

Table IV
Computer Data for the F-4E

Computer	Lear Siegler ARN-101
Function(s)	Navigation, Fire Control
Acquisition Date	Not operational yet
Memory Size	48,000 (original 32,000)
Bits/Word	16
Program Size at Acquisition	31,195
Present Program Size	41,802
Original Cycle Time	78.1% requirements
Present Cycle Time	80.9% requirements
Computer	Digital LRU-1
Function(s)	Radar
Acquisition Date	Not available
Memory Size	16,000
Bits/Word	16
Program Size at Acquisition	about 8,000
Present Program Size	about 10,000
Original Cycle Time	Not available
Present Cycle Time	Not available

(From Ref 15)

The growth in occupied memory, thus far, for the ARN-101 software has been about 35 percent for both the F-4E and the RF-4C systems. The growth in available memory, though, will soon be 100 percent. The growth in timing is a little bit different for the two systems. The timing for the F-4E system has grown only about 2 percent while the timing for the RF-4C system has grown about 40 percent. One of the reasons for the large difference in increased

Table V
Computer Data for the RF-4C

Computer	Lear Siegler LS-52 ARN-101
Function(s)	Navigation, Fire Control
Acquisition Date	Not operational yet
Memory Size	48,000 (original 32,000)
Bits/Word	16
Program Size at Acquisition	27,328
Present Program Size	37,213
Original Cycle Time	56.3% requirements
Present Cycle Time	80.0% requirements

(From Ref 15)

timing is the spare requirement levied for when the systems become operational. The F-4E system timing was already near the 80 percent level and was not allowed to grow, while the RF-4C timing was allowed to grow to the 80 percent level from a 56.3 percent level. Timing was controlled through software techniques (Ref 25).

Very little has occurred to the Digital LRU-1 system aboard the F-4E, as compared to the ARN-101 system. The size of the software has grown about 25 percent and timing has not been considered a problem. No spare requirements were levied for this computer system when it became operational (Ref 15).

The F-15

The data in Table VI shows that the F-15 has three computers with software on-board. The first is the IBM

Table VI
Computer Data for the F-15

Computer	IBM AP-1
Function(s)	Navigation, Fire Control
Acquisition Date	Nov 74
Memory Size	16,000
Bits/Word	32
Program Size at Acquisition	10,500
Present Program Size	about 13,000
Original Cycle Time	50% requirements
Present Cycle Time	about the same
Computer	HCM-230
Function(s)	Radar
Acquisition Date	Nov 74
Memory Size	16,000
Bits/Word	24
Program Size at Acquisition	about 16,000
Present Program Size	about 16,000
Original Cycle Time	Not available
Present Cycle Time	Not available
Computer	Texas Instruments TI-2520
Function(s)	Electronic Warfare
Acquisition Date	Fall 1976
Memory Size	32,000
Bits/Word	16
Program Size at Acquisition	16,000-20.000
Present Program Size	about the same
Original Cycle Time	Not available
Present Cycle Time	Not available

(From Ref 20)

AP-1. The software for this computer has grown about 25 percent. The computer system had requirements for 25 percent spare memory and 40 percent spare timing capability when it became operational. Plans have been made to modify the computer for an add-on capability to double the size of its memory. This is the maximum number of locations that the central processor can address directly, and represents a 100 percent growth in available memory (Ref 20).

The second computer listed in Table VI is the HCM-230. There were no spare requirements levied on this system when it became operational. As a result, the software occupied essentially all of the available memory at that time. Parts of the programs have been rewritten several times to gain available memory. Some of the modes of operation have had to be deleted to make room for additional requirements. There are plans to change to a computer with a memory of 24,000 words in the summer of 1978. Long-range plans are to expand the memory to 96,000 words. The central processor can direct address only 48,000 locations, but bank switching would be used to address the remaining. This expansion would represent a growth in available memory of 500 percent (Ref 20)!

Bank switching is one of the techniques used to address more memory than the central processor is capable of addressing directly. For example, if a computer can direct address 16,000 words, two "banks" of memory (each 16,000 words) could be provided. With the use of a special instruction, the central processor would "switch" to one bank or the other. Then, only 16,000 addresses need be used, but each location would contain different information depending on which memory bank is accessed (Ref 20).

The TI-2520 computer had an original memory with 16,000 words and was expanded to the present 32,000 words to meet operational specifications. This represents a growth in available memory of 100 percent again. Only minor changes have been made to the software in the last several months as the first major update has not, yet, been made. No spare requirements were levied on this system, either (Ref 20).

Major Farmer, at the Warner Robins ALC, noted that a list had been compiled of capabilities desired now for the F-15 in the future. If all of these capabilities were incorporated into the system, the software would require a computer with four times the present memory. A hard line approach to change is used to keep the software within present capabilities.

The E-3A (AWACS)

The E-3A computer systems will become operational between September 1977 and October 1978. This data is not useful for historical analysis in this report, but is included here for two reasons. First, the IBM CC-1 system, which will become operational in October 1978, is already having 16,000 words added to its memory. The second reason is that indications are all four systems will utilize nearly all available memory when they become operational (Ref 32). The data for the E-3A is shown in Table VII.

Table VII
Computer Data for the E-3A

Computer	IBM CC-1
Function(s)	System Maintenance
Acquisition Date	Oct 1978
Memory Size	112,000 Core
Bits/Word	32
Program Size at Acquisition (anticipated)	6400 BYTES
Present Program Size	Not applicable
Original Cycle Time (anticipated)	10 seconds
Present Cycle Time	Not applicable
Computer	Delco M-311 (ASN-119)
Function(s)	Navigation (Inertial)
Acquisition Date	Sept 1977
Memory Size	6,000
Bits/Word	12
Program Size at Acquisition	6,000
Present Program Size	Same as above
Original Cycle Time	Not available
Present Cycle Time	Not available
Computer	Northrop NDC-1070
Function(s)	Navigation
Acquisition Date	Sept 1977
Memory Size	16,000
Bits/Word	16
Program Size at Acquisition	16,000
Present Program Size	Same as above
Original Cycle Time	Not available
Present Cycle Time	Not available

"Table VII continued"

(From Ref 32)

Table VII
Computer Data for the E-3A
"continued"

Computer	Westinghouse MX AN/AYK-8
Function(s)	Radar Data Correlator
Acquisition Date	Sept 1977
Memory Size	104,000
Bits/Word	18
Program Size at Acquisition	104,000
Present Program Size	Same as above
Original Cycle Time	Not available
Present Cycle Time	Not available

(From Ref 32)

Core space is considered critical by the personnel at the Oklahoma City ALC, even before the E-3A is operational. When the system becomes operational, plans will have to be made to expand the memory on each computer. Expansion rework will be very costly (Ref 32).

The F-111D, F-111F, and FB-111A

The software for the F-111D, F-111F, and FB-111A computer systems is maintained individually. That is, the software for the computers aboard the F-111D is maintained separately from the software for the computers on-board the F-111F or the FB-111A. Each aircraft has three computers with software on-board. One IBM CP-2 computer is referred to as the General Navigation Computer (GNC). A second IBM CP-2 computer is referred to as the Weapons Delivery Computer (WDC). The third computer is an Auto .ics D26J-41.

Few changes have been made to the software for this last computer and little information is available on it (Ref 13). The Autonetics computer will not be discussed. This data is summarized in Table VIII.

The current tape version of the software on the F-111D is D-18. The two tape versions prior to D-18 included 24 changes for optimization and 8 changes for word savers in the D-16 version of the OFP and 23 changes for optimization and word savers in the D-17 tape versions. The current software tape version included 2 changes for word savers and a major module optimization resulting in 100 changes. At this point, the GNC has only 179 words available (1.1%) and the WDC has only 262 words available (1.6%) (Ref 13).

The latest software tape update for the F-111F is F-12 and included 27 changes. Prior to that, the F-11 OFP contained 23 changes. These changes included word savers, optimization, changes in users' requirements, and correction of problem areas. Several modules were rewritten, clarified, and optimized, but they still perform the same functions. Currently, there are 1,130 words available in the GNC (6.9%' and 1,316 words available in the WDC (8.0%) on the F-111F (Ref 13).

At the beginning of every cycle on the FB-111A OFP, it has been necessary to delete and/or optimize certain routines to make room for higher priority routines.

Table VIII
Computer Data for the F-111D, F-111F, FB-111A

Computer	IBM CP-2 (4 PI)
Function(s)	General Navigation & Backup Weapons Delivery if WDC fails
Acquisition Date	Spring 1973
Memory Size	16,896
Bits/Word	18
Program Size at Acquisition	Not Available
Present Program Size	about 15,766 F-111F about 16,475 FB-111A about 16,717 F-111D
Original Cycle Time	Not available
Present Cycle Time	Not available
Computer	IBM CP-2 (4 PI)
Function(s)	Weapons Delivery, Self test, & Backup General Navigation
Acquisition Date	Spring 1973
Memory Size	16,896
Bits/Word	18
Program Size at Acquisition	Not available
Present Program Size	about 15,580 F-111F about 16,876 FB-111A about 16,634 F-111D
Original Cycle Time	Not available
Present Cycle Time	Not available
Computer	Autonetics D26J-41
Function(s)	Control and Logic
Acquisition Date	Spring 1973
Memory Size	4096
Bits/Word	12
Programs Size at Acquisition	Not available
Present Program Size	4056
No timing information available	

(From Ref 13)

Unfortunately, the words saved by these deletions were not fully documented. However, before the start of the FB-15 cycle, approximately 1,200 words were saved by deleting 16 routines in the GNC and the WDC. Even with that, there remain only 421 words available in the GNC (2.6%) and 20 words available in the WDC (0.1%) (Ref 13).

The IBM CP-2 computer will be modified to an IBM CP-2A in the future. Module Core Memory will contain 32,000 words, expandable to 64,000 words. This will represent a growth in available memory of nearly 300 percent (Ref 13).

From Table VIII, one can see that the information about timing was listed as "Not available." The information was provided, but not as a single value. For worst case situations, the percentage of available time used was provided for each of seven different rate groups on each aircraft. The highest percentages of available time used on the three aircraft ranged from 90 percent to about 97 percent. This is pointed out because timing is considered a problem by the personnel at the Sacramento ALC. The modified IBM CP-2A will have a speed three times faster than the IBM CP-2. This represents a growth in timing capability of 200 percent (Ref 13).

The SRAM, Minute Man I, II, and III Missiles

The data for the SRAM Missile is shown in Table IX. Data for the Minute Man I, Minute Man II, and Minute

Man III missiles are shown in Tables X-XII. Memory in the SRAM Missile was full from the beginning. Growth in memory for the Minute Man missiles has been between 10 and 20 percent; the memories being nearly full at present. Growth for these systems is very small because

Table IX
Computer Data for the SRAM Missile AGM-69A

Computer	Delco M-301 CP-908/A
Function(s)	Navigation
Acquisition Date	July 1974
Memory Size	2,000
Bits/Word	8
Program Size at Acquisition	2,019 from program listing - full
Present Program Size	Same as above
Original Cycle Time	Not available
Present Cycle Time	Not available

(From Ref 32)

Table X
Computer Data for the Minute Man I

Computer	Autonetics D-37A
Function(s)	Guidance & Control
Acquisition Date	about 1960
Memory Size	3500
Bits/Word	24
Program Size at Acquisition	about 3150 (90%)
Present Program Size	about 3500 (100%)
Original Cycle Time	100% capability
Present Cycle Time	Same as above

(From Ref 21)

Table XI
Computer Data for the Minute Man II

Computer	Autonetics D-37C
Function(s)	Guidance & Control
Acquisition Date	about 1966
Memory Size	7,000
Bits/Word	24
Program Size at Acquisition	80-85%
Present Program Size	about 100%
Original Cycle Time	100% capability
Present Cycle Time	Same as above

(From Ref 21)

Table XII
Computer Data for the Minute Man III

Computer	Autonetics D-37D
Function(s)	Guidance & Control
Acquisition Date	about 1970
Memory Size	14,000
Bits/Word	24
Program Size at Acquisition	about 80%
Present Program Size	90-95%
Original Cycle Time	100% capability
Present Cycle Time	Same as above

(From Ref 21)

the systems were designed for just one purpose, and few changes are expected to the guidance and control functions of a missile (Refs 32: 21).

The SRAM Missile Carriers

The SRAM Missile can be launched from two different aircraft, the B-52 and the FB-111. The same computer system is used for launching in both aircraft. The computer is an Autonetics D26J-103H. Data on this system is contained in Table XIII.

Once again, memory for this system has been essentially full from the operational date of the system. No information was obtained on changes to the software. For SRAM use, the OFP and diagnostics are both resident in core. In future plans, the diagnostics will have to be scrubbed to make room for other material (Ref 32). Actual growth of the system cannot be measured under these circumstances.

Radar Warning Systems

Data was obtained on two Radar Warning Systems used on various aircraft. The data is shown in Table XIV.

Table XIII
Computer Data for the SRAM Carrier
Aircraft, B-52 and FB-111

Computer	Autonetics D26J-103H
Function(s)	OFP
Acquisition Date	July 1974
Memory Size	16,000
Bits/Word	24
Program Size at Acquisition	about 16,000
Present Program Size	about 16,000
Original Cycle Time	Not available
Present Cycle Time	Not available

(Ref 32)

The first system is the ROLM 1601. THis system had an original memory of 5,700 words, which was expanded to 12,000 words by a modification. That modification also made the computer speed three times as fast. The growth in available memory is just over 100 percent. The software has grown from about 3,700 words to about 10,500 words, representing a growth of about 185 percent (Ref 7).

The second system in Radar Warning computers is the Dalmo-Victor DVP-25. This system was acquired just this

Table XIV
Computer Data for Radar Warning Computers
Aboard Various Aircraft

Computer	Rolm 1601
Function(s)	Radar Warning
Acquisition Date	April 1973
Memory Size	12,000
Bits/Word	16
Program Size at Acquisition	about 3700
Present Program Size	about 10,500
Original Cycle Time	Not available
Present Cycle Time	Not available
Computer	Dalmo-Victor DVP-25
Function(s)	Radar Warning & Power Management
Acquisition Date	April 1977
Memory Size	16,000
Program Size at Acquisition	about 10,500
Present Program Size	about 16,000
Original Cycle Time	Not available
Present Cycle Time	Not available

(From Ref 7)

year and has already used all available memory. There is a pending modification which will expand the memory to 24,000 words. When this modification is accomplished, a growth in available memory of 50 percent will have been made. The software has grown from 10,500 words to about 16,000 words. This is a growth in the software of just over 50 percent, also (Ref 7).

IV Analysis of the Memory and Timing Parameters for Various Airborne Computer Systems

This chapter provides a comparative analysis of the data presented in Chapter III. The first part of the chapter is an analysis of the spare memory parameter for various airborne computer systems. The analysis looks at the actual growth in software size and the actual or planned growth in memory size (hardware). Then, the growth of computer systems is examined within general functional areas. The second part of the chapter is an analysis of the spare timing parameter for the various airborne computer systems. It examines the problems encountered with timing in avionics computers.

Spare Memory in Airborne Computer Systems

The growth in software will be examined first in this part of the analysis. Data concerning the actual growth in software was available for fifteen of the twenty-five computers included in Chapter III. These computers include one aboard the F-106, three on the C-5A, one on the A-7D, two aboard the F-4E and RF-4C, two on the F-15, four aboard missiles, and the two Radar Warning systems. The actual growth in software for these computers is shown in Figure 9.

At first glance, it looks like more than half of the computer systems have growth in software of less than 25 percent. From Figure 9, one would be inclined to recommend 50 percent or less spare memory. But a closer look at the data is necessary.

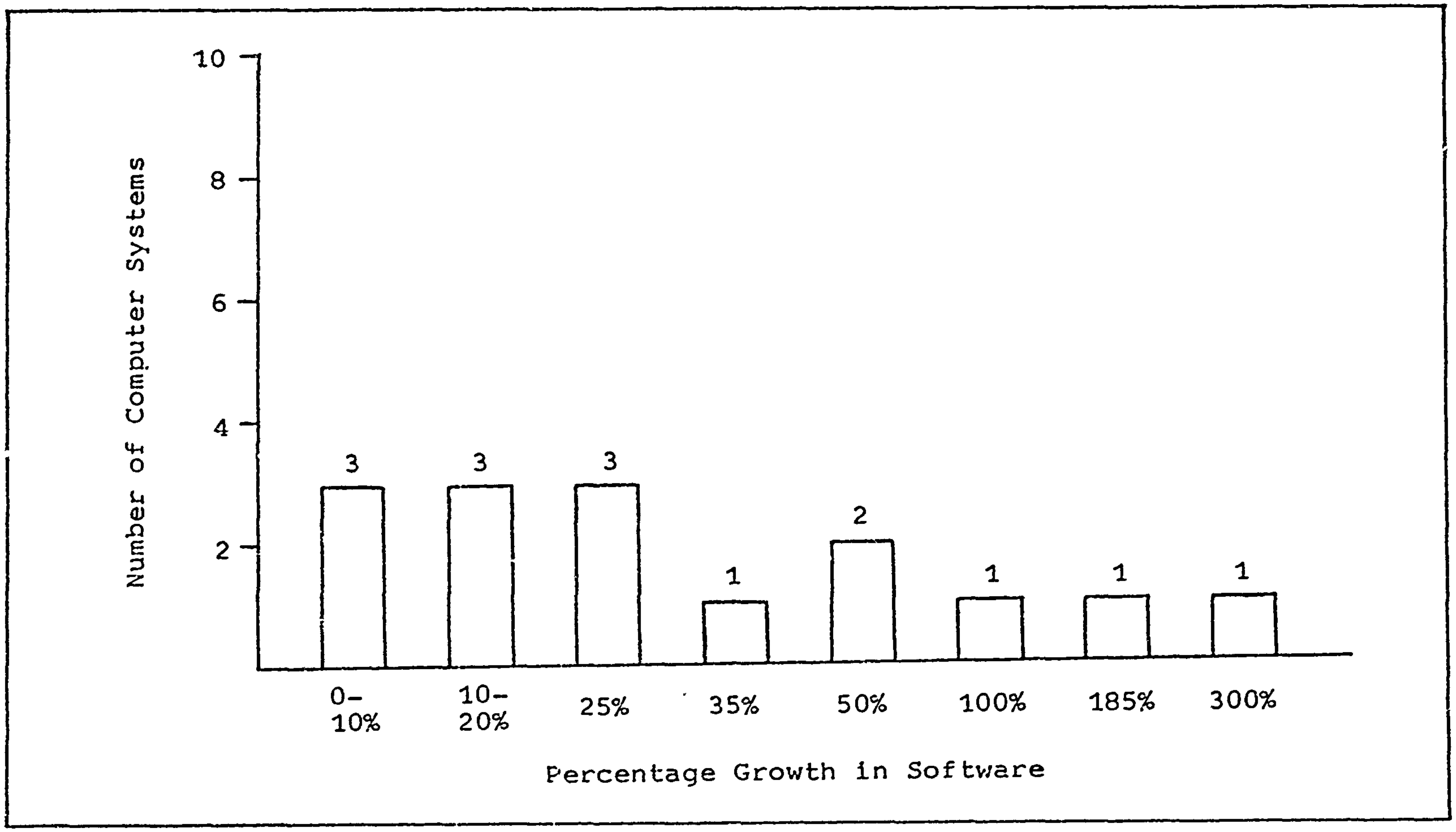

Figure 9. Actual Software Growth in 15 Airborne Computer Systems

Of the nine computer systems with 25 percent or less growth in software, seven are at 100 percent capacity, presently, and cannot grow unless memory is expanded in the future. Of these nine, one is the status monitoring system and four are missile systems where little growth is anticipated. Of the remaining four in these nine, three will be expanded to allow for the further growth of the software. Of the six computer systems with more than 25 percent growth in software, four of these will also be expanded to allow for even greater software growth. Over the expected life cycle of the majority of these systems it appears that actual growth in software will be much greater than 25 percent.

The computer systems were then grouped according to general functions. The functional classes used are missile guidance, status monitoring, navigation, navigation and weapons control, radar, electronic warfare, and general purpose computers. The data for the fifteen systems in Figure 9 is shown in Table XV. The range of actual software growth and the number of systems within each functional group are listed.

Although the data in Table XV is not conclusive, it does show important trends. Computer systems for missile guidance and status monitoring have had very small growth in software. Computer systems for the remaining general functions have a wide range of software growth; that is,

Table XV
Actual Software Growth for 15 Computer
Systems According to General Functions

Function	Number of Computers	Percentage Growth
Missile Guidance	4	0 - 20%
Status Monitoring	1	10%
Navigation	2	50 - 100%
Navigation & Weapons Control	3	25 - 35%
Radar	2	0 - 25%
Electronic Warfare	2	50 - 85%
General Purpose	1	300%

between 0 and 300 percent. Eight of those ten systems have experienced between 25 and 100 percent actual growth in software.

This study was originally designed to examine the actual growth in software on various avionics computers. During the data collection and data analysis, it became apparent that an examination of the actual or planned growth in memory is also necessary. This growth is the amount of memory that has been or will be added to the physical computer. Since many of the computer systems included in this report are relatively new, anticipated growth of the software may be more important than the actual growth observed thus far.

Data for actual or planned growth in memory was available for twenty airborne computer systems. These systems include one computer aboard the F-106, three on the C-5A, one on the A-7D, two aboard the F-4E and RF-4C, three on the F-15, three on the F-111 series aircraft, five aboard missiles, and the two Radar Warning systems. The actual or planned growth in memory for these twenty systems in shown in Figure 10.

Once again, it appears from Figure 10 that almost half of the computer systems have not and will not have memory added to the computer. These computer systems were also grouped according to general functions, using the same functional classes as previously. That data is presented in Table XVI in the same format as Table XV.

As before, one can see a trend when the data is presented in the form of Table XVI. Of the nine computers with no hardware growth, five are the missile systems and one is the status monitoring system. That leaves only three other systems with no hardware growth; but in two of those systems, software has grown 50-100 percent. Excluding the missile systems and the status monitoring system, only one of the remaining systems has had negligible software growth and has no present plans for expansion of its memory.

Looking at the bottom fourteen computer systems in Table XVI, which includes the primary computer functions aboard aircraft, one can see that nine of those systems have or will have expanded memory 100-300 percent. That is approximately 65 percent of the aircraft computers

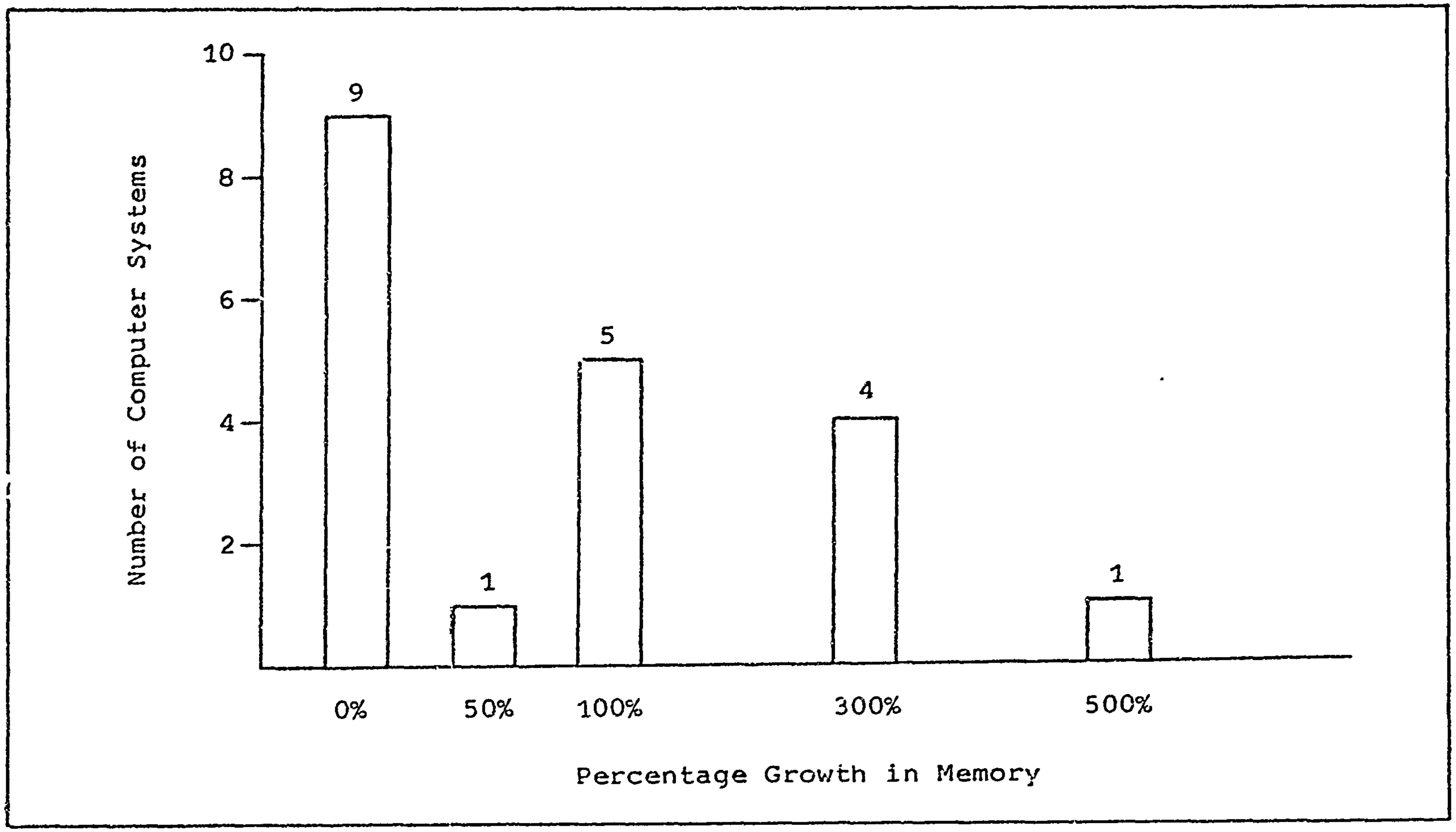

Figure 10. Actual or Planned Growth of Memory in 20 Airborne Computer Systems

studied with primary functions. This shows an anticipated growth much higher than the actual growth in software, thus far, for these particular computer systems.

The 100-300 percent range for hardware growth may seem large, but it really represents only a small number of discrete values. Because of the binary construction and operation of a computer, expansion of memory will usually be to a power of two. The 100 percent increase represents a memory twice as large as the original. The 300 percent increase doubles the memory again so that the memory is four times the original size. These are the most common values for memory expansion, though not the only ones. Through the use of bank switching, as explained earlier, memory could be tripled if three banks of memory were used, two being added to the original memory. As a result, there is only a small range of values between the 100 percent and 300 percent growth in memory.

Table XVI
Actual or Planned Memory Growth for 20 Computer Systems According to General Functions

Function	Number of Computers	Percentage Growth
Missile Guidance	5	0%
Status Monitoring	1	0%
Navigation	2	0%
Navigation & Weapons Control	6	100 - 300%
Radar	2	0 - 500%
Electronic Warfare	3	50 - 100%
General Purpose	1	325%

As was stated in Chapter II, Dr. Boehm's study recommended that 50-100 percent or more spare memory capability be provided in new computer systems because of cost considerations. For primary aircraft functions, the data in this report indicate that 100-300 percent additional memory is required during the life cycle of these systems and should be recommended. As shown, this result is supported by Dr. Boehm's cost study. For other systems, such as a missile system or a status monitoring system, only about 25 percent spare memory capability is necessary.

To support this conclusion, an example is used to give one an idea of the costs involved. The discussion for the A-7D in Chapter III stated that it will cost about $100,000 per aircraft to replace the TC-2 on-board computer with a new TC-2A. This would double the available memory, expanding from 16,000 words to 32,000 words. Mr. Larry Lang, at the F-15 SPO, stated that there is a pending Value Engineering Proposal (VECP) there that will allow the memory size in the HCM-230 computer on the F-15 to be doubled in the remaining aircraft to be built. This would also expand the memory from 16,000 words to 32,000 words. This modification will cost $5,000 per aircraft. To modify the computer in those aircraft already built would cost more. This author realizes that the technological costs for each computer may be different. Also, it is not known how much more a computer with a larger memory would have cost originally for the A-7D. But there is a significant cost difference for the two systems, and this should cause one to question the present methods used in acquiring additional memory for avionics computers. It appears that it costs much less to provide the additional capability at the start, rather than trying to gain it later through modifications.

Spare Timing in Airborne Computer Systems

Timing in avionics computers cannot be analyzed in the same way as the memory problem. From the data, one can see that timing information concerning growth was available on only one aircraft computer system, that being the ARN-101 system on the F-4. As explained in Chapter III, that system has a spare timing requirement causing this information to be maintained. Timing information on the F-111 was provided, but this was not growth data. Also, the speed of four computer systems was tripled, but these were results of hardware modifications designed primarily to provide more memory.

With several of the systems, personnel responsible for the software maintenance stated that timing was considered a problem but could be controlled through software techniques. Still other personnel said that it was not a problem for the same reason.

There seems to be little data available and little general agreement on the problem because few people understand it. Spare timing is required by ASD/ENAIA, but a usable definition of this parameter does not exist.

Therefore, personnel must try to guess exactly what is desired. To satisfy the requirement on the F-4 system, the contractor demonstrated that the software contained an approximate number of instructions in each class. Then, based on theoretical time of execution for each instruction and an estimated number of loops for parts of the software, the theoretical cycle time was computed and shown to be within the limits set.

The obscurity surrounding the timing problem is caused by two factors. First, there is no clear definition nor understanding as to what exactly should be included in the timing figures. How much input data should be assumed? How many loops should be counted when loops in the software are timed? There is little guidance available in this area of concern. Second, there is presently no method available to actually time the software during execution in avionics computers. Because of the limited memory size in avionics computers, priority functions of the software do not include calculating and maintaining cycle time of the programs. Therefore, cycle time is not provided in airborne computers. How can one time the software, then? Theoretical cycle time can be calculated, but what guarantee is there that the computer is actually executing instructions at that speed?

Until these questions are solved, definitions stated, and methods provided to accomplish the desired results, timing will continue to be a problem. The results of this report cannot provide a value to be recommended for timing.

The 25 percent figure presently required by ASD/ENAIA appears to be adequate. Regardless of the function of the computer, the problems with timing were basically the same. One figure for spare timing seems to be satisfactory for all avionics computers.

The need for spare timing may not be as great as the need for spare memory. A study should be made to determine additional costs required to work around timing constraints during software maintenance. The results of such a study could demonstrate the need for spare timing, or the lack of such a need. If a need does exist, then adequate spare timing parameters can be provided.

V Conclusions and Recommendations

This report has attempted to present an analysis of the requirements for spare memory and spare timing capability in avionics computers. Other studies have been made relating the costs of memory to the cost of developing and maintaining software. But this author believes that the amount of spare memory or spare timing capability provided should be directly related to the amount of either parameter required operationally during the life cycle of the system.

This author realizes that during weapons system development in a tight economy, many cost trade-offs are made. To keep the immediate costs as low as possible, additional memory is often not acquired during the development of the computer system. As studies referred to earlier in this report noted, in many cases, almost no extra memory may be provided.

Of course, one may argue that the additional cost may be a waste of money if the system does not grow to capacity. It is true, in that situation, that some money would have been spent unnecessarily. But the savings that could be realized from the majority of the systems would far exceed the loss on the few systems that would not grow to capacity during their life cycles.

Costs of the hardware were not a subject of this research effort. But it may be very beneficial for another study to examine this specific issue. Cost data could be gathered concerning the cost of a system at a given size, the cost of a larger system at the same time, and the cost to expand the first system to the size of the second system at a later date. The results of such a study, considered in light of likely system expansion, could have important implications for minimizing life cycle costs.

To overcome some of the problems experienced with hardware and software in avionics computers, a system to collect and maintain data concerning memory and timing needs to be established. This research effort has identified the lack of information available in these areas. Much of the little data that does exist is insufficient for analysis because of the trade-offs involved among memory, timing, and the cost of the software during software maintenance. With more memory or spare timing, programming changes are easier. With less memory or spare timing, programming changes are more difficult, causing increased personnel costs. Consequently, not knowing how to calculate the costs of these trade-offs, there is little attempt to record data concerning these parameters.

This report had two objectives. First, to propose and defend a standard optimum software parameter choice for each of the two parameters discussed. And second, to attempt to identify feasible parameter range choices for various avionics software functions. The objectives were accomplished for only one of the two parameters, i.e., spare memory. The problems associated with spare timing were discussed in detail. This study demonstrated the need for greater spare memory or add-on capability. Future studies could show the most economical way to satisfy this need.

To summarize, the following is a list of conclusions and recommendations resulting from this research effort:

1. Inadequate spare memory is being provided with new avionics computer systems.
2. Confusion exists in the Air Force and among contractors concerning spare timing requirements levied on Air Force weapons systems.
3. 100-300 percent spare memory or add-on capability should be provided for avionics computers that process data for navigation, weapons control, radar, electronic warfare, or any other function that has changing mission requirements.
4. 25 percent spare memory or add-on capability should be provided for avionics computers associated with missiles, status monitoring, fault isolation, or similar functions.
5. Not enough data is available concerning timing in avionics computers; no conclusion can be reached concerning the ASD/ENAIA requirement for 25 percent spare timing.
6. A definition and guidelines should be established for timing in avionics computers if spare timing is to be required.

7. A study to determine additional costs incurred in working around timing constraints during software maintenance should be made.

8. A study of a comparative analysis of the costs involved with obtaining more memory originally versus modifying for expansion at a later date should be made.

Bibliography

1. *A Computer Centralization Cost Model for Conceptual Design*. The RAND Corporation, R-1268-PR, September 1973. (AD 776 028)

2. *Air Force Command and Control Information Processing in the 1980s*. RAND Corporation, June 1974. (ADA 017 128)

3. *Aircraft Avionics (Digital Avionics Study Report), Volume 1, Final Report*, April 1973. (AD 912 998)

4. *Aircraft Avionics Trade-Off Study (AATOS)*. McDonnell Aircraft Company, ASD/XR-73-20, November 1973. (AD 915 874)

5. Anderson, Christine M. *Aerospace Higher Order Language Processing*, Technical Report AFAL-TR-73-151. Wright-Patterson Air Force Base, Ohio: Air Force Avionics Laboratory, June 1973. (AD 912 922)

6. *Avionic Computer Software Design and Development Engineering Practices*. EXHIBIT ASD/ENAIA 76-1. Wright-Patterson Air Force Base, Ohio: Aeronautical Systems Division, AFSC, 31 August 1976.

7. Black, Joe (personal correspondence). Warner-Robins Air Force Base, Georgia: Warner-Robins ALC/MMECE, 26 September 1977.

8. Boehm, Barry W. *Some Information Processing Implications of Air Force Space Missions: 1970-1980*. RAND Corporation, RM-6213-PR, January 1970. (AD 703 279)

9. Boehm, Barry W. and Allen C. Haile. *Information Processing/Data Automation Implications of Air Force Command and Control Requirements in the 1980s (CCIP-85 Executive Summary*. SAMSO, Los Angeles, California, SAMSO/TR-72-122, February 1972.

10. *DOD Weapon Systems Software Management Study: Appendix A, Findings and Recommendations of Previous Studies*. Laurel, Maryland: The John Hopkins University, Applied Physics Laboratory, June 1975. (ADA 031 598)

11. Furtaw, R. N. *Aircraft Avionics Tradeoff Study*. Canoga Park, California: Hughes Aircraft Company, ASD/SR-73-19, November 1973. (AD 917 204)

12. Gates, Howard P. Jr. *Electronics-X: A Study of Military Electronics with Particular Reference to Cost and Reliability. Volume 2: Complete Report*. Institute for Defense Analyses, January 1974. (ADA 001 065)

13. Green, Robert (personal correspondence). McClellan Air Force Base, California: Sacramento ALC/MMEC, 20 September 1977.

14. Hinton, Edward S. "Operational Flight Software." Proceedings of the Aeronautical Systems Software Work-Shop. Air Force Systems Command, July 1974. (ADA 004 547)

15. Hougaard, Hugh (telephone interview). Hill Air Force Base, Utah: Ogden ALC/MMEC. 8 September 1977.

16. Information Processing/Data Automation Implications of Air Force Command and Control Requirements in the 1980s (CCIP-85), Volume V. SAMSO, TR-72-122, January 1973. (AD 907 626)

17. Kirkham, Edward (telephone interview). Kelly Air Force Base, Texas: San Antonio ALC/MMEC. 1 September 1977.

18. Kossiakoff, A. DOD Weapon Systems Software Management Study. Laurel, Maryland: The John Hopkins University, Applied Physics Laboratory, June 1975. (ADA 022 160)

19. Kravec, John M. Reprocurement Data for Avionic Digital Computer Software. Technical Report ASD-TR-74-9, February 1974. (AD 921 529)

20. Lang, Larry (telephone interview). Wright-Patterson Air Force Base, Ohio: F-15 SPO/YFEA. 9 September 1977.

21. Lewis, Marv (telephone interview). Hill Air Force Base, Utah: Ogden ALC/MMECM. 14 September 1977.

22. Management Guide to Avionics Software Acquisition, Volume I. Logicon, Technical Report ASD-TR-76-11, Volume I, June 1976. (ADA 030 591)

23. Management Guide to Avionics Software Acquisition, Volume II. Logicon, Techical Report ASD-TR-76-11, Volume II, June 1976. (ADA 030 592)

24. Management Guide to Avionics Software Acquisition, Volume IV. Logicon, Technical Report ASD-TR-76-11, Volume IV, June 1976. (ADA 030 594)

25. McDonald, Glenn (telephone interview). Hill Air Force Base, Utah: Ogden ALC/MMEC. 8 September 1977.

26. Pontius, Harry E. Life Cycle Guidelines for Weapon System Software Management. Fort Belvoir, Virginia: Defense Systems Management School, May 1976. (ADA 029 376)

27. Proceedings of a Symposium on the High Cost of Software Held at the Naval Postgraduate School, Monterey, California, September 17-19, 1973. (AD 777 121)

28. Reliability and Integrity of Large Computer Programs. California University, March 1974. (AD 779 339)

29. Schmidt, Jerry D. "AFLC and User Requirements for Avionic Software." Proceedings of the Aeronautical Systems Software Workshop. Air Force Systems Command, July 1974. (ADA 004 547)

30. Schultz, Kenneth (telephone interview). Wright-Patterson Air Force Base, Ohio: ASD/ENAIA. 8 September 1977.

31. Squire, Jon S. "Low Cost Operational Flight Software." Proceedings of the Aeronautical Systems Software Workshop. Air Force Systems Command, July 1974. (ADA 004 547)

32. Statham, Phil (personal correspondence). Tinker Air Force Base, Oklahoma: Oklahoma City ALC/MMECT. 7 September 1977.

33. Summary Notes of a Government/Industry Software Sizing and Costing Workshop. Electronics Systems Division, AFSC, October 1974. (ADA 026 964)

34. Thayer, Richard H. Rome Air Development Center R & D Program in Computer Language Controls and Software Engineering Techniques. RADC-TR-74-80, April 1974. (AD 778 836)

35. Trainor, W. Lynn. "Trends in Avionic Software." Proceedings of the Aeronautical Systems Software Workshop. Air Force Systems Command, July 1974. (ADA 004 547)

36. Vendt, Bruce A. Software Support for the F-16 Avionic Computers. Wright-Patterson Air Force Base, Ohio: Air Force Institute of Technology. December 1975. (ADA 020 361)

APPENDIX A

DEFINITIONS

APPENDIX A

Definitions

Automatic Test Equipment: Electronic devices capable of automatically or semi-automatically measuring selected parameters of an electronic, mechnical, or electromechanical item being tested and making a comparison to accept or reject these measured values in accordance with predetermined limits.

Automatic Test Equipment Software: The computer programs used to control test operations and the procedures required to test various hardware systems and subsystems using test stimuli and comparing the system response with predetermined acceptance parameters.

Avionics: All electronics on-board an air vehicle.

Bank Switching: A technique used to address more memory than the central processor is capable of addressing directly by using a special instruction to switch the central processor from one bank of memory to another, both having the same address range.

Bit: A binary digit representing either 0 or a 1. Several bits are used together to represent a computer word.

Computer: A physical system consisting of one or more processors, one or more stoage devices, and data input/output facilities.

Computer Program: A collection of instructions in a formal language called a programming language that describes a process.

Computer Speed: The computational capability of a processor stated in terms of millions of instructions per second (MIPS) or thousand of operations per second (KOPS) based on a measured average of execution times of certain mixes of various instructions.

Data: Physical phenomena chosen by convention to represent certain aspects of our conceptual and real world. Data are used to transmit, store, and retrieve information and to derive new information by manipulating the data according to formal rules.

Execution: A term which applies to the phenomena in which a processor performs an operation described by a program instruction which the processor retrieves from a store component of the respective computer.

Firmware: Programs contained in read-only memories (ROM), electrically-alterable ROMs (EROM), or programmable ROMs (PROM) that cannot easily be altered by the user.

Operational Flight Programs: All computer programs executed in an airborne system.

Simulation Software: That set of computer programs and data developed to operate in a digital computer in support of specific crew training devices.

Software: A collection of computer programs and the data processed by them. Frequently, software is defined to also include the associated documentation.

Software Documentation: All written material necessary to describe, understand, and modify software. It can be manuals, flow charts, computer language description, program description, subprogram interface description, program listings, machine codes, core utilization charts, timing sequences, data description, mathematical algorithms, or any other similar documentation.

Software Maintenance: Modification or change in the program instructions for various reasons during the operational phase of a computer system.

Software Reliability: 1) The number of software errors inherent in the computer programs. 2) The probability that a run of the program will give the desired output with a valid set of input data.

Software Validation: The process by which the developer tests a given program to assure that the product complies with the Test Requirement Document or engineering source data.

Software Verification: The operational evaluation by which the program is tested and proved to be adequate for operational application and compatible with its associated hardware.

Timing: The cycle time of the programs; i.e., the length of time it takes for the entire program to execute, which varies with the amount of data to be processed and the number of loops to be made.

Value Engineering Change Proposal: A proposed engineering change initiated by the contractor during the development or production of an item. The contractor is rewarded by a share of either the instant savings, the future acquisition savings, or the collateral savings.

APPENDIX B

PERSONNEL INTERVIEWED

APPENDIX B

Personnel Interviewed

The following personnel were interviewed either in person or by telephone during the research documented in the body of this report.

HQ AIR FORCE LOGISTICS COMMAND

Mr. Mark van den Broek	LOAK
Captain Nicholas Babiak	LOAK

AIR LOGISTICS CENTERS

Mr. Charles Singleton	Warner Robins ALC
Mr. Joe Black	Warner Robins ALC
Mr. Dave Corder	Oklahoma City ALC
Mr. Phil Statham	Oklahoma City ALC
Mr. Robert Green	Sacramento ALC
Mr. Ed Kirkham	San Antonio ALC
Mr. Dave Thornell	Ogden ALC
Mr. Glenn McDonald	Ogden ALC
Mr. Hugh Hougaard	Ogden ALC
Mr. David Erickson	Ogden ALC
Mr. Marv Lewis	Ogden ALC

AERONAUTICAL SYSTEMS DIVISION (AFSC)

Lieutenant Colonel Larry Taylor	ENAIA
Major Ken Schultz	ENAIA
Mr. Larry Lang	F-15 SPO
Major Radford	F-16 APO
Mr. Murchlan	A-10 SPO

APPENDIX C

SAMPLE DATA FORM

APPENDIX C

Sample Data Form

AIRCRAFT TYPE: ______________________________

ON-BOARD COMPUTER TYPE: ______________________________

WEIGHT: ______________________________

FUNCTION(S): ______________________________

OPERATIONAL DELIVERY DATE OF
COMPUTER AND SOFTWARE: ______________________________

MEMORY SIZE: ______________________________

BITS PER WORD: ______________________________

TIMING CAPACITY OR SPEED: ______________________________

PROGRAM SIZE AT OPS DELIVERY: ______________________________

PRESENT PROGRAM SIZE: ______________________________

PROGRAM CYCLE TIME AT OPS DELIVERY: ______________________________

PRESENT PROGRAM CYCLE TIME: ______________________________

OTHER COMMENTS:

Vita

Gary Brian Wigle was born on [redacted] in [redacted]. He graduated from high school in [redacted] 1969 and from the United States Air Force Academy in 1973, receiving a'Bachelor of Science degree in Physics and a commission in the USAF. After serving three years as a Computer Programmer/Systems Analyst with the 20th Air Dividion Headquarters at Fort Lee Air Force Station, Virginia, he entered the Air Force Institute of Technology (Systems Management program) in September 1976.

Permanent Address: [redacted]